

 **Object Constraint Language (OCL)**

By Arif Khan

Arif.Khan@cs.ucl.ac.uk

 **Object Constraint Language ((OCL)**

< 55 SECOND VIDEO CLIP TO GO IN HERE – ATTACHING IT WOULD HAVE MADE THE FILE TOO LARGE. VIDEO CLIP IS FROM KNIGHT RIDER THE GAME, AND THIS WILL BE USED FOR EXAMPLES DURING THE PRESENTATION >


 **Object Constraint Language ((OCL)**

Why Is It Needed?

- Graphics model not descriptive enough
- Natural Language leads to ambiguity


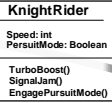
Why OCL?



- Less Mathematical than other languages
- Allows more content to be added to the basic UML
- Makes the model:
 - Precise
 - Consistent
 - Complete


 **Object Constraint Language ((OCL)**


What is OCL?

Terms we know:

Object →  

Constraint →  

Language → 

 **Object Constraint Language ((OCL)**


What is OCL?

- Means of extending the UML
- Allows us to specify 4 Types of constraints on an object:
- An Invariant, a post condition, a pre-condition and a guard (more on these later)

OCL

```

graph TD
    OCL --- ExpressionLanguage[Expression Language]
    OCL --- ModellingLanguage[Modelling Language]
    OCL --- FormalLanguage[Formal Language]
  
```

 **Object Constraint Language ((OCL)**

Properties of OCL?

Expression:

- Guaranteed to occur without side effect
- Cannot change the model in any way
- Just evaluates and returns a value

Modelling:

- Not programming
- Not possible to define a logical flow
- An expression cannot change the state of the system

Formal:

- All constructs are defined
- Only 1 unique reading of the
- Expression possible



Object Constraint Language ((OCL))

Types of Constraints:

Invariant

- Constraint that applies to **ALL** instances of class (or type or interface)
- An expression that evaluates to true if the condition is met.
- **All** Invariants must **ALWAYS** evaluate to true.

Precondition

- Must be true at the moment the operation is to be executed.

Post Condition

- Evaluate to true at the moment the operation ends

Guard

- Must be true before state transition can occur



Object Constraint Language ((OCL))

Context

- Need to specify which item from the model it's a constraint on
- Usually a group of UML diagrams
- For an invariant the constraint applies to all instances of that class or type or interface
- The keyword "self" is used to specify the context of the constraint



Object Constraint Language ((OCL))

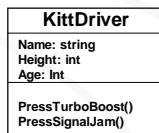
Invariant on Attributes:

Constraint: DRIVER MUST BE OVER 18



Context KittDriver inv:
Age > = 18

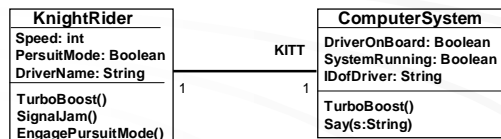
Context KittDriver inv:
Height > = 150



Object Constraint Language ((OCL))

Invariant on Associations

- Constraint can be placed on attributes that are connected over associations



Context KnightRider inv:
DriverName = KITT.IDofDriver



Object Constraint Language ((OCL))

Invariant on Collections:

- Many associations are not 1 to 1.
- To deal with this we treat them as collections and apply **SET OPERATIONS**

SET OPERATIONS

- includes (object) – for membership test
- union (set of objects)
- intersection (set of objects)
- notEmpty (set of objects)

LOGIC OPERATIONS

- Exists, forAll
- XOR, AND, – ∃ Weapons (∃ WeaponController)



Object Constraint Language ((OCL))

Conditions on Associations:

EnemyTracker

pre: not Enemy -> includes(e)

post: Enemy = Enemy @ pre -> including (e)



How would this be done without OCL?

- Add note in UML, but OCL is formal way

Object Constraint Language (OCL)

Guards

- The UML contains **STATE TRANSITION DIAGRAMS**
- Guards specify conditions that need to be met before a transition can occur
- Evaluated before the transition so can be thought of as a pre-condition

Object Constraint Language (OCL)

Guards

```

stateDiagram-v2
    state NormalMode
    state PursuitMode
    NormalMode --> PursuitMode : EngagePursuiteMode() (self.Operational)
    PursuitMode --> NormalMode : EndPursuitMode()
  
```

- Boolean condition checked before state transition.
- Example of a guard

Object Constraint Language (OCL)

Constraints and Inheritance

- Subclass can strengthen constraint but not relax it
- Encapsulation and abstraction
- Increase down inheritance hierarchy

Object Constraint Language (OCL)

Broken Constraints

- What to do if constraint is broken is not specified
- OCL is only a modelling language
- Action cannot be carried out as a result
- Kleppe and Warmer proposed including ACTIONS in OCL at the UML 2000 conference.

Object Constraint Language (OCL)

Further Reading

- Big area, a whole formal language
- Area of research and development
- Submission been made for OCL 2.0 as part of UML 2.0
- Look at book for more information

Object Constraint Language (OCL)

Summary

- Adding constraints allows us to express more in an object model
- OCL provides a formal, non-ambiguous way of doing this
- Invariants, Pre Condition, Post Condition and Guards are employed
- Extensions to these are being worked on
- OCL 2.0 part of UML 2.0



Object Constraint Language (OCL)

Questions?

ASK →

